

## CLAIMS

What is claimed is:

1. A method for managing communication flow between a user interface and a computer application performing a task comprising a plurality of steps, the sequence of the steps controlled by the application and the progression through the steps controlled by a user operating the user interface, comprising:

(a) mapping each step to an output file containing information to be sent to the user interface in support of the step;

(b) mapping each task to a output generator that generates an output sent via an output medium to the user interface based upon the content of the output file; and

(c) instantiating a task object modeling the task, the task object receiving a progressional input from the user interface and receiving a step sequence input from the application, the task object further performing the steps of

(i) comparing the progressional input to the step sequence input to identify a subsequent step,

(ii) identifying the output file mapped to the subsequent step, and

(iii) calling the output generator mapped to the task to generate an output to the user interface based upon the content of the output file mapped to the subsequent step.

2. The method of claim 1 wherein each step is mapped to an output file on a one to one basis.

3. The method of claim 2 wherein each task is mapped to a output generator on a one to one basis.

4. The method of claim 3 wherein the output generator is a Java servlet generating a  
5 hypertext markup language (HTML) output.

5. The method of claim 3 wherein the output generator is a Java applet generating a hypertext markup language (HTML) output.

6. The method of claim 4 wherein the HTML output is generated from static HTML files having dynamic content inserted therein by an HTML template.

7. The method of claim 4 wherein the output medium is hypertext transport protocol (HTTP).

8. The method of claim 3 wherein the output medium is an interactive voice response (IVR) protocol.

9. The method of claim 4 wherein each step is mapped to an output file using a screen map.

10. The method of claim 1 wherein the task object models the task through a subclass of step objects, each step object representing a step within the task and having a method to perform the step.

11. The method of claim 6 wherein the progressional input comprises an instruction to proceed to the next step, go back to the previous step, or cancel the step.

5 12. The method of claim 7 wherein the step sequence input from the application is received by implementing an abstract method that passes step sequence information from the application to the task object.

13. The method of claim 8 wherein step (c)(i) is performed by a utility object.

10 14. The method of claim 9 wherein the output generator that generates an output further comprises a method to generate an error message as an output to the user interface.

15 15. The method of claim 9 wherein the method to generate an error message further comprises extending an exception handling class.

16. The method of claim 1 wherein the sequence of the steps controlled by the application is obtained by the application from an external source.

20 17. The method of claim 16 wherein the external source is an initialization file.

18. The method of claim 16 wherein the external source is a database.

19. A framework for managing communication flow between a user interface and a computer application performing a task comprising a plurality of steps, the sequence of the steps controlled by the application and the progression through the steps controlled by a user operating the user interface, comprising:

5 (a) an initialization file mapping each step to an output file containing information to be sent to the user interface in support of the step and mapping each task to a output generator that generates an output sent via an output medium to the user interface based upon the content of the output file; and

(b) a task object modeling the task, the task object receiving a progressional input from the user interface and receiving a step sequence input from the application, the task object further comprising methods to perform the steps of

(i) comparing the progressional input to the step sequence input to identify a subsequent step,

(ii) identifying the output file mapped to the subsequent step, and

15 (iii) calling the output generator mapped to the task to generate an output to the user interface based upon the content of the output file mapped to the subsequent step.